

BAB II

KAJIAN TEORI

Pada bab II ini dijelaskan mengenai beberapa teori tentang penjadwalan, penjadwalan kuliah, metode penyelesaian penyusunan jadwal kuliah, algoritma genetika, dan algoritma memetika yang akan digunakan sebagai landasan dalam pembahasan selanjutnya.

A. Penjadwalan

Menurut Kamus Besar Bahasa Indonesia, jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan secara terperinci sedangkan penjadwalan adalah proses untuk menyusun jadwal atau memasukkan kegiatan ke dalam jadwal. Masalah penjadwalan ini dapat ditemui dalam kehidupan sehari-hari di berbagai bidang kegiatan, seperti di lingkungan sekolah, universitas, organisasi, di lingkungan instansi seperti rumah sakit, di lingkungan yang berkaitan dengan sarana transportasi seperti penjadwalan keberangkatan bus, kereta api, kapal, dan pesawat serta di lingkungan lainnya.

Menurut Sahid (1997: 4), dalam kehidupan sehari-hari kegiatan dapat dikelompokkan menjadi 3 kategori: kegiatan yang bersifat tidak disengaja (*accidental events*), sementara (*temporary events*), dan kegiatan rutin (*regular activity*). Kegiatan bersifat tidak disengaja mungkin terjadi tanpa rencana atau jadwal yang tidak diketahui seseorang sebelumnya. Kegiatan sementara mungkin terjadi pada satu kesempatan, seperti pertemuan resmi para pemimpin antar negara, pertunjukan

musik, atau dapat juga dalam interval waktu beberapa hari atau minggu seperti kompetisi olahraga, sesi ujian dan lain sebagainya. Kegiatan reguler adalah kegiatan yang biasa dikerjakan secara berkala baik harian, mingguan, bulanan atau dalam interval waktu dalam beberapa jam. Contoh khusus kegiatan reguler seperti keberangkatan kereta api, bus, atau pesawat, program acara televisi maupun radio, pertemuan antara guru dengan murid di sekolah, dan lainnya. Kegiatan yang bersifat berkala ini membutuhkan suatu rencana yang menjelaskan mengenai waktu dan tempat dari masing-masing kegiatan.

Adanya jadwal serangkaian kegiatan bermanfaat untuk orang-orang yang terlibat langsung maupun tidak dalam suatu acara sehingga dapat mengelola waktu dengan baik (Sahid, 1997). Siswa di sekolah maupun mahasiswa di universitas membutuhkan jadwal untuk mengetahui apa yang harus dipelajari sedangkan seorang guru atau dosen perlu mempersiapkan materi yang harus diajarkan. Terdapat banyak contoh lain mengenai jadwal dalam kehidupan sehari-hari yang sering kita jumpai seperti jadwal pertandingan sepak bola, jadwal pameran kesenian, jadwal acara program televisi dan sebagainya.

B. Penjadwalan Universitas

Pada universitas, terdapat dua kegiatan utama yaitu kuliah dan ujian yang melibatkan mahasiswa, dosen, dan juga beberapa komponen lain. Penyusunan jadwal kuliah maupun ujian merupakan masalah yang kompleks karena banyak hal yang harus diperhitungkan. Kuliah berlangsung setiap semester pada setiap tahun akademik. Masing-masing kegiatan tersebut, selain dihadiri kelompok mahasiswa dan

dosen juga melibatkan sumber daya tertentu seperti ruang kelas, atau laboratorium dan sebagainya. Untuk menentukan aktivitas akan berlangsung dalam slot waktu dan menggunakan ruang tertentu, jadwal diperlukan agar semua kegiatan berjalan tanpa konflik yang melibatkan mahasiswa, dosen maupun ruang. Suatu jadwal kuliah biasanya disusun sebelum dimulainya tahun ajaran baru sebelum kuliah dimulai sedangkan suatu jadwal ujian disusun sebelum waktu ujian dimulai. Suatu jadwal kuliah bersifat berkala dengan periode satu minggu. Berbeda dengan jadwal kuliah, jadwal ujian disusun tidak berkala. Sesi ujian hanya berlangsung selama periode waktu tertentu, misalnya dua atau satu minggu.

Belum terdapat istilah baku dalam penjadwalan yang telah disepakati secara umum oleh para peneliti, meskipun permasalahan ini telah menjadi bidang studi yang luas selama puluhan tahun. Penulis yang berbeda mungkin menggunakan istilah yang berbeda untuk konsep yang sama atau istilah yang sama namun dalam arti berbeda. Sebagai contoh, beberapa menggunakan istilah fakultas untuk dosen yang mengajar di universitas. Selain itu, beberapa penulis juga menggunakan istilah kelas untuk menggambarkan sekelompok mahasiswa yang mengambil beberapa mata kuliah, sementara penulis lain menggunakan istilah tersebut dengan makna sekelompok mahasiswa yang mengambil kombinasi mata kuliah yang sama. Daftar istilah dasar yang berkaitan dengan masalah penjadwalan di universitas menurut Sahid (1997: 30) dan Abdullah (2006: 9) adalah sebagai berikut.

Tabel 2.1: Daftar istilah dasar pada penjadwalan universitas

Istilah	Definisi
Kegiatan	Suatu kegiatan yang akan dijadwalkan. Dalam hal ini adalah ujian dan kuliah.
Kuliah	Salah satu bagian dari program pendidikan atau kurikulum yang ditawarkan oleh suatu lembaga.
Ujian	Kegiatan mahasiswa harus menyelesaikan sekumpulan pertanyaan atau masalah sebagai evaluasi dari mata kuliah yang diikuti.
Slot waktu	Interval waktu event dapat dijadwalkan.
Individu	Peserta yang mengikuti kegiatan perkuliahan atau ujian yaitu mahasiswa dan dosen
Kendala	Persyaratan atau ketentuan yang harus dipenuhi dalam jadwal.
Konflik	Suatu pelanggaran terhadap kendala terdapat dua kegiatan kuliah maupun ujian dalam waktu atau ruang yang sama.

C. Penjadwalan Kuliah

Suatu jadwal kuliah terdiri atas kombinasi mata kuliah, slot waktu dan ruang dengan setiap perkuliahan melibatkan sekelompok mahasiswa dan dosen yang bertatap muka selama sejumlah periode waktu untuk membahas mata kuliah tertentu. Penjadwalan kuliah melibatkan penentuan mahasiswa dan dosen yang akan berpartisipasi dalam sesi pengajaran tertentu dalam subjek tertentu dan penugasan setiap sesi pengajaran pada ruang dan periode waktu (Sahid, 1997: 44).

Setelah masalah penjadwalan didefinisikan dengan baik, biasanya memuat beberapa tujuan yang harus dicapai dan kendala yang harus terpenuhi. Di sisi lain, suatu masalah penjadwalan seringkali didefinisikan oleh sekumpulan persyaratan,

beberapa diantaranya menjelaskan tentang tujuan penjadwalan dan yang lainnya adalah kendala. Eiselt dan Laporta (1987) mengklasifikasikan kendala masalah penjadwalan kuliah ini dalam dua kriteria yaitu kendala *hard* dan kendala *soft*. Kendala *hard* merupakan kendala yang harus terpenuhi dalam penyusunan jadwal agar diperoleh jadwal yang layak, sedangkan kendala *soft* merupakan kendala yang menentukan kualitas jadwal yang dihasilkan.

Kendala *hard* berdasarkan permasalahan penjadwalan secara umum bertujuan untuk membentuk jadwal yang layak sehingga harus memenuhi kendala-kendala berikut (Rossidoria dkk, 2006: 406).

1. Hanya terdapat satu kuliah di masing-masing ruangan pada slot waktu yang sama.
2. Setiap ruangan harus cukup untuk menampung semua mahasiswa yang mengikuti kuliah serta memenuhi semua fasilitas yang dibutuhkan.
3. Tidak terdapat mahasiswa yang mengikuti 2 atau lebih kuliah pada slot waktu yang sama.

Setelah memenuhi seluruh kendala *hard*, penyusunan jadwal dilanjutkan dengan meminimumkan fungsi penalti kendala *soft* yaitu dengan cara meminimalisasi pelanggaran terhadap kendala *soft*. Kendala *soft* yang telah didefinisikan dalam permasalahan penjadwalan universitas secara umum menurut Rossidoria dkk (2006: 406) adalah sebagai berikut.

1. Mahasiswa tidak mengikuti kuliah pada slot waktu terakhir setiap harinya.

2. Mahasiswa tidak hanya mengikuti satu kuliah dalam 1 hari.
3. Mahasiswa tidak mengikuti kuliah lebih dari 6 jam berturut-turut.

D. Prosedur Penyusunan Jadwal Kuliah

Masing-masing lembaga pendidikan perlu untuk menyusun jadwal setiap waktu, semester atau tahun. Pada universitas, jadwal biasanya disusun oleh bagian administrasi akademik dari fakultas namun terdapat beberapa lembaga yang menyusun jadwal secara masing-masing jurusan. Berdasarkan permasalahan yang harus diselesaikan, menurut Sahid (1997: 30) masalah penjadwalan secara umum adalah untuk mencari solusi masalah berikut.

1. Penentuan daftar mata kuliah yang akan ditawarkan dan jumlah kelas serta banyaknya tatap muka masing-masing mata kuliah.
2. Penugasan dosen untuk mengajar tiap-tiap mata kuliah yang sesuai dengan bidangnya.
3. Penempatan mata kuliah, ruang dan slot waktu serta dosen untuk perkuliahan.
4. Penempatan masing-masing mahasiswa dengan mata kuliah dan kelas berdasarkan permintaan mahasiswa sehingga meminimalkan terjadinya bentrokan.

Menurut Sahid (1997: 30), prosedur penyusunan jadwal secara umum dapat digambarkan pada tahapan berikut.

1. Mengumpulkan data yang terdiri atas.

- a. daftar mata kuliah atau ujian yang akan dijadwalkan, daftar ruang beserta kapasitas juga ketersediannya, serta periode waktu termasuk durasi perkuliahan yang dialokasikan untuk penjadwalan;
 - b. informasi mengenai ketersediaan dosen yang bertanggung jawab mengampu tiap-tiap mata kuliah;
 - c. informasi mengenai data mahasiswa dan matakuliah yang diikuti.
2. Merumuskan masalah dan metode penyelesaiannya yang meliputi penggunaan model atau pendekatan tertentu, solusi manual, metode heuristik dan sebagainya,
 3. Mengaplikasikan metode penyelesaiannya untuk merumuskan masalah,
 4. Mencetak dan mempublikasikan jadwal yang dihasilkan,
 5. Menyimpan jadwal dalam *database* yang mudah diakses untuk perbaikan kedepannya.

E. Metode Penyelesaian Masalah Penjadwalan Kuliah

Berbagai metode untuk masalah umum optimisasi sudah diterapkan untuk menyelesaikan berbagai macam masalah penjadwalan. Metode-metode tersebut termasuk pewarnaan graf berbasis heuristik, *simulated annealing*, *tabu search*, algoritma genetika, pemrograman logika kendala (*constraint logic programming*), jaringan syaraf tiruan (*neural networks*) (Sahid, 1997: 60). Selain beberapa metode tersebut, algoritma memetika juga dapat menjadi salah satu pilihan metode penyelesaian dalam penjadwalan kuliah. Beberapa metode akan dijelaskan dalam sub-bab berikut.

1. Algoritma

Algoritma adalah urutan logis langkah-langkah penyelesaian masalah yang disusun secara sistematis (Rinaldi Munir, 2005: 176).

Menurut Donald E. Knuth (Suarga, 2012: 2), algoritma memiliki beberapa ciri sebagai berikut.

1. Algoritma mempunyai awal dan akhir, suatu algoritma harus berhenti setelah mengerjakan serangkaian tugas. Dengan kata lain, suatu algoritma memiliki langkah terbatas.
2. Setiap langkah pada algoritma harus didefinisikan dengan tepat sehingga tidak memiliki arti ganda.
3. Algoritma memiliki masukan (*input*) atau kondisi awal.
4. Algoritma memiliki keluaran (*output*) atau kondisi akhir.
5. Algoritma harus efektif, bila diikuti benar-benar maka akan menyelesaikan persoalan.

Berdasarkan ciri algoritma yang diuraikan di atas, sifat utama suatu algoritma (Suarga, 2012: 4) adalah sebagai berikut.

1. *Input*

Suatu algoritma memiliki *input* sebelum dilaksanakan, dapat berupa nilai-nilai peubah yang diambil dari himpunan khusus.

2. *Output*

Suatu algoritma akan menghasilkan *output* setelah dilaksanakan, atau algoritma akan mengubah kondisi awal menjadi kondisi akhir, dimana nilai *output* diperoleh dari nilai input yang telah diproses melalui algoritma.

3. *Definiteness*

Langkah-langkah yang ditulis dalam algoritma terdefinisi dengan jelas sehingga mudah dilaksanakan oleh pengguna algoritma.

4. *Finiteness*

Suatu algoritma harus memberi *output* setelah sejumlah langkah dilakukan terhadap setiap *input* yang diberikan.

5. *Effectiveness*

Setiap langkah dalam algoritma dapat dilaksanakan dalam suatu selang waktu tertentu sehingga pada akhirnya diperoleh solusi sesuai yang diharapkan.

6. *Generality*

Langkah-langkah algoritma berlaku untuk setiap himpunan *input* yang sesuai dengan persoalan yang diberikan, tidak hanya untuk himpunan tertentu.

Algoritma berfungsi untuk mempermudah kerja atau memudahkan kita untuk menyusun suatu program. Selain itu, algoritma juga dapat mengatasi masalah logika dan matematika termasuk dalam masalah penjadwalan. Pemilihan algoritma yang tepat dalam penyusunan jadwal berpengaruh dengan hasil dari jadwal itu sendiri.

2. Algoritma Genetika

Menurut Sahid (1997: 64), algoritma genetika merupakan algoritma evolusioner untuk memecahkan masalah optimisasi berdasarkan pada ide-ide evolusi seleksi alam. Algoritma ini mensimulasikan proses evolusi dengan menghasilkan populasi solusi layak dan menerapkan beberapa operator genetik untuk menghasilkan populasi baru berdasarkan aturan seleksi. Menurut Kralev (2009: 291), ide mengenai algoritma evolusioner diperkenalkan pertama kalinya oleh Rechenberg pada tahun 1973 sedangkan ide mengenai algoritma genetika diajukan oleh John Holland dari Universitas Michigan pada tahun 1960-an. John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi dapat diformulasikan dengan terminologi genetika. Algoritma genetika dikembangkan lebih lanjut oleh John Holland bersama murid-murid dan rekan kerjanya dari Universitas Michigan pada tahun 1960-an dan 1970-an. Sejak itu algoritma genetika telah dipelajari, diteliti dan diaplikasikan secara luas di berbagai bidang. Pada awal tahun 1992 pemrograman mengenai genetika mulai diperkenalkan oleh J.H. Koza (Kralev, 2009: 291).

Algoritma genetika menggunakan analogi secara langsung dari kebiasaan yang alami yaitu seleksi alam. Dalam ilmu biologi, sekumpulan individu yang sama, hidup dan berkembang bersama dalam suatu area, disebut dengan populasi. Algoritma genetika bekerja dengan suatu populasi yang terdiri atas individu-individu, yang masing-masing individu mempresentasikan solusi yang mungkin bagi suatu permasalahan. Algoritma genetika memiliki performansi yang baik untuk masalah-masalah selain optimisasi kompleks dari satu variabel atau multivariabel. Salah satu

aplikasi algoritma genetika adalah pada permasalahan optimisasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi, seperti optimisasi penjadwalan mata kuliah.

Pada algoritma genetika, kandidat solusi (yang selanjutnya disebut ‘solusi’) dari suatu masalah direpresentasikan sebagai kromosom. Kromosom ini merupakan representasi dari solusi yang ada. Kumpulan kromosom disebut sebagai populasi. Masing-masing kromosom pada populasi akan dievaluasi menggunakan fungsi *fitness*, yaitu fungsi yang mengukur secara kuantitatif suatu kromosom untuk bertahan pada populasi. Pembentukan populasi baru dilakukan dengan menerapkan operator-operator genetika secara iteratif sampai dipenuhi kriteria berhenti, misalnya diperoleh populasi yang baik dari populasi sebelumnya. Operator dasar dalam algoritma genetika adalah seleksi, *crossover* dan mutasi. Operator genetik *crossover* membentuk dua solusi (orang tua) dan menggabungkan dua solusi tersebut setelah satu atau lebih solusi (anak) dihasilkan. Orang tua yang dipilih merupakan orang tua dari semua solusi dalam populasi saat ini Kravev (2009: 291).

Algoritma genetika bekerja dengan menggunakan pendekatan random, sehingga nilai-nilai yang dihasilkan adalah nilai random. Pada kasus penjadwalan dengan model genetika, semakin banyak iterasi yang dilakukan maka waktu yang dibutuhkan akan semakin lama.

3. *Local Search* (Pencarian Lokal)

Menurut Sahid (1997: 61), salah satu prosedur perbaikan yang paling sederhana adalah algoritma pencarian lokal. Algoritma pencarian lokal adalah teknik

perbaikan berulang yang dapat diterapkan secara umum untuk setiap masalah optimisasi dengan ruang solusi X yang baik, suatu fungsi objektif $f(s)$ terdefinisi untuk setiap solusi layak s elemen X , dan persekitaran $N(s)$ untuk setiap solusi s . Diberikan sebuah solusi awal, algoritma akan menghasilkan urutan solusi dengan mencari solusi baru secara berulang di persekitaran solusi saat ini yang menjadi solusi baru saat ini. Tetangga yang terpilih diterima jika memiliki kualitas yang lebih baik daripada solusi saat ini. Algoritma dapat menerima solusi awal dengan kualitas yang lebih baik atau solusi di persekitaran yang memberikan peningkatan kualitas solusi terbesar, disebut *steepest descent* atau *steepest ascent*. Algoritma pencarian lokal berakhir ketika tidak terdapat solusi di persekitaran solusi saat ini yang berkualitas lebih baik.

4. Algoritma Memetika

Algoritma memetika pertama kali diperkenalkan oleh Moscato & Norman pada tahun 1992. Moscato dan Norman menggambarkan sebuah algoritma evolusioner yang menggunakan teknik pencarian lokal. Menurut Kralev (2009: 291), ide Moscato dan Norman kemudian dirumuskan oleh Radcliffe dan Surrey yang membuat perbandingan antara algoritma genetika dan memetika. Dalam algoritma memetika, meme dianggap sebagai unit informasi yang dapat mereplikasi diri melalui komunikasi antar individu, tanpa ada keterkaitan dengan tingkat replikasi dari gen masing-masing individu dan dapat disebarkan ke individu lain melalui cara seperti pencarian lokal, pindah silang, evolusi atau dengan cara lainnya. Proses replikasi

mime disebut juga *process of self-reproduction* atau *the memetic life-cycle* yang dilanjutkan dengan penyebaran mime. Proses replikasi dari orang tua kepada anaknya hanya terjadi pada gen (*vertical transmission*) yang merupakan konsep dari algoritma genetika, sedangkan proses replikasi pada mime dapat juga terjadi dalam satu individu (*horizontal transmission*). Gabungan kedua konsep itu merupakan dasar algoritma memetika, sedangkan mime dalam algoritma memetika disebut dengan gen.

Algoritma memetika didasari oleh teori evolusi biologi neo darwinian dan pendapat Dawkin tentang mime sebagai unit evolusi kultural yang mampu melakukan perbaikan terhadap dirinya sendiri. Algoritma memetika adalah suatu metode pencarian heuristik yang mengkombinasikan antara Algoritma genetika dengan metode pencarian lokal, yang secara bersama-sama dapat meningkatkan kualitas pencarian solusi (Moscato, 1989). Perbedaan algoritma genetika dan algoritma memetika menurut Kralev (2009: 292) terletak pada solusi yang dihasilkan. Dasar dikembangkannya algoritma genetika adalah menghasilkan sebuah solusi berdasarkan konstruktif heuristik. Tujuan utama algoritma ini adalah diberikan kegiatan awal sehingga kegiatan tersebut ditempatkan pada jadwal mingguan. Berbeda dengan algoritma genetika, algoritma memetika menghasilkan sebuah solusi berdasar pada pencarian lokal. Tujuan utama algoritma memetika ini adalah dengan diberikan di awal suatu peraturan dalam kegiatan untuk ditempatkan pada jadwal mingguan sedemikian rupa sehingga dapat menghasilkan solusi dengan nilai terbaik akan solusi yang dihasilkan.

Algoritma memetika dapat digambarkan dalam diagram alir seperti ditunjukkan pada gambar 2.1.



Gambar 2.1 Diagram alir algoritma memetika

Struktur dari algoritma memetika dapat didefinisikan dengan langkah-langkah berikut.

1. Pengkodean

Dalam penggunaan algoritma genetika maupun memetika cara untuk merepresenatsikan solusi ke dalam kromosom adalah dengan pengkodean. Dalam pengkodean kromosom setiap solusi harus dikodekan dan kromosom harus merepresentasikan tepat satu solusi. Secara umum, suatu kromosom berbentuk

$$\text{kromosom} = (gen_1, gen_2, \dots, gen_n)$$

n adalah jumlah gen dalam suatu kromosom dan setiap gen diisi oleh nilai dari gen.

Menurut Suyanto (2005: 7) terdapat 3 skema yang paling umum digunakan yaitu.

a. *Real number encoding*

yaitu pengkodean nilai gen berada dalam interval $[0,R]$ dimana R adalah bilangan positif yang biasanya bernilai 1.

b. *Discrete decimal encoding*

yaitu pengkodean nilai gen dapat bernilai dari salah satu bilangan bulat dalam interval $[0,N]$.

c. *Binary encoding*

yaitu setiap gen hanya dapat bernilai 0 atau 1.

Berikut merupakan gambaran skema pengkodean gen-gen dalam kromosom.

x1			x2			x3			
0,3150			1,000			0,0257			Real number encoding
gen 1			gen 2			gen 3			

7	3	4	9	15	28	6	0	1	Discrete decimal encoding
gen 1	gen 2	gen 3	gen 4	gen 5	gen 6	gen 7	gen 8	gen 9	

1	0	0	1	1	1	0	0	0	Binary encoding
gen 1	gen 2	gen 3	gen 4	gen 5	gen 6	gen 7	gen 8	gen 9	

Gambar 2.2 Pengkodean kromosom dalam algoritma memetika

Pada gambar 2.2 terdapat tiga variabel, yaitu x_1 , x_2 , x_3 , yang dikodekan ke dalam suatu kromosom yang terdiri atas 3 gen (untuk *real number encoding*) sedangkan pada *discrete decimal encoding* maupun *binary encoding*, ketiga variabel x_1, x_2, x_3 dikodekan ke dalam kromosom yang terdiri atas 9 gen (masing-masing variabel dikodekan dalam 3 gen) (Suyanto, 2005).

2. Pembentukan populasi awal

Algoritma memetika merupakan algoritma heuristik yang bekerja pada populasi, yaitu kumpulan kromosom atau solusi yang akan diperbaharui pada setiap generasinya. Pembentukan pupolasi awal yang berisi kumpulan kromosom sebanyak ukuran populasi biasanya dilakukan secara acak.

3. Fungsi *fitness*

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi akan bertahan hidup sedangkan yang bernilai *fitness* rendah akan mati. Pada permasalahan optimisasi, jika solusi yang dicari adalah memaksimalkan suatu fungsi h , maka nilai *fitness* yang digunakan adalah nilai dari fungsi h tersebut, yaitu $f = h$ dengan f adalah fungsi *fitness* namun jika masalahnya adalah meminimalkan fungsi h , maka fungsi h tidak dapat digunakan secara langsung. Oleh karena itu, fungsi *fitness* yang dapat digunakan adalah $f = \frac{1}{h}$, semakin kecil nilai dari fungsi h maka semakin besar nilai f . namun hal ini akan menjadi masalah jika h bernilai 0 yang mengakibatkan f bernilai tak hingga. Untuk mengatasinya, h perlu ditambah suatu bilangan yang dianggap sangat kecil sehingga nilai *fitness*nya menjadi

$$f = \frac{1}{(h + a)},$$

dengan a adalah bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan (Suyanto, 2005).

4. Fungsi penalti

Fungsi penalti memiliki nilai suatu kromosom yang menentukan kualitas atau kelayakan solusi masalah. Nilai penalti menunjukkan bobot pelanggaran terhadap suatu kendala atau batasan. Dalam masalah optimisasi, nilai penalti biasanya diminimumkan sehingga diperoleh solusi dengan jumlah pelanggaran kendala yang paling sedikit. Semakin kecil nilai penalti maka akan semakin meningkatkan kualitas solusi, sedangkan kelayakan solusi ditentukan oleh kendala suatu nilai penalti.

5. Seleksi orang tua

Seleksi orang tua merupakan bagian evolusi yaitu proses mendekati kandidat solusi permasalahan yang diharapkan. Seleksi orang tua adalah operator algoritma memetik yang dilakukan pada setiap generasi untuk memilih kromosom-kromosom dari populasi yang akan menjadi orang tua bagi generasi berikutnya. Menurut Sri Kusumadewi (2006: 235), langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu tempat seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif individu tersebut terhadap semua individu dalam tempat seleksi tersebut.

Proses seleksi dapat dilakukan dengan beberapa metode antara lain *roulette wheel selection*, *tournament selection*, *rank selection*, *truncation selection*. Penjelasan mengenai metode-metode tersebut adalah sebagai berikut.

a. *Roulette wheel selection*

Dalam *roulette wheel selection*, masing-masing kromosom menempati potongan lingkaran pada roulette wheel dengan ukuran yang proporsional sebanding

dengan nilai *fitness*nya. Semakin besar nilai *fitness*, maka semakin besar kemungkinan kromosom itu untuk terpilih. Langkah awal dari *roulette wheel selection* adalah menghitung nilai *fitness* masing-masing kromosom. Setelah itu dihitung ukuran masing-masing kromosom berdasarkan perbandingan probabilitas antara nilai *fitness* setiap kromosom dengan total nilai *fitness*. Langkah selanjutnya adalah membangkitkan bilangan real secara random antara 0 dan 1 untuk menentukan kromosom yang bertahan hidup dan menjadi induk.

b. *Tournament selection*

Pada *tournament selection*, sekelompok kecil kromosom diambil secara random, kemudian dipilih satu kromosom yang memiliki nilai *fitness* paling tinggi untuk menjadi orang tua. Hal ini dilakukan berulang-ulang hingga didapatkan jumlah orang tua yang diinginkan.

c. *Rank selection*

Metode *rank selection* bekerja dengan terlebih dahulu meranking atau mengurutkan kromosom di dalam populasi berdasarkan *fitness*-nya. Kemudian memberi nilai *fitness* baru berdasarkan urutannya. Kromosom dengan *fitness* terburuk akan memiliki *fitness* baru yang bernilai 1, terburuk kedua bernilai 2, dan seterusnya. Sehingga kromosom yang memiliki *fitness* terbaik akan memiliki nilai *fitness* N , dengan N adalah jumlah kromosom dalam sebuah populasi. Setelah proses pengurutan dan pemberian nilai *fitness* baru, setiap kromosom akan memiliki

kesempatan yang lebih kecil untuk terpilih, sehingga metode ini menyebabkan konvergensi menjadi lambat (Obitko, 1998).

d. *Truncation selection*

Truncation selection biasa digunakan oleh populasi yang jumlahnya besar. Semua kromosom dalam populasi diurutkan berdasarkan nilai *fitness*-nya kemudian dipilih sejumlah kromosom terbaik yang diseleksi sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang trunk yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk. Individu yang berada di bawah nilai ambang tidak akan menghasilkan keturunan (Sri Kusumadewi, 2004).

6. Pindah silang (*crossover*)

Pindah silang (*crossover*) merupakan komponen paling penting dalam algoritma genetika dan memetika karena proses inilah yang dapat mendekati kromosom-kromosom sebagai kandidat solusi yang diharapkan. Kromosom yang mengarah pada nilai penalti yang lebih bagus diperoleh dari proses memindahsilangkan dua buah kromosom yang diperoleh dari seleksi orang tua. Pindah silang dapat menghasilkan satu atau beberapa kromosom yang biasa disebut kromosom anak.

Pindah silang dapat berakibat buruk jika jumlah populasinya sangat kecil karena suatu kromosom dengan gen-gen yang mengarah ke solusi (kromosom yang diharapkan) akan sangat cepat menyebar ke kromosom-kromosom lainnya. Untuk mengatasi masalah tersebut pindah silang hanya dapat dilakukan untuk probabilitas

tertentu misal p_c , artinya pindah silang dilakukan jika suatu bilangan random $[0,1)$ yang dibangkitkan kurang dari p_c yang ditentukan. Pada umumnya, p_c ditentukan mendekati satu, misal 0,8.

Terdapat beberapa jenis operator *crossover* antara lain yaitu *one point crossover*, *two point crossover*, dan *uniform crossover*. Penjelasan mengenai operator *crossover* tersebut adalah sebagai berikut.

a. *One point crossover*

Menurut Suyanto (2005: 14), pindah silang yang paling sederhana adalah *one point crossover*. Dalam *one point crossover* satu titik dipilih secara random untuk membagi orang tua yang pertama, kemudian gen pertama hingga titik yang terpilih diturunkan ke kromosom anak sesuai posisinya, sedangkan orang tua yang lainnya diperiksa, jika terdapat nilai gen yang belum ada pada anak, maka ditambahkan ke kromosom anak dengan cara *left-to-right*.

b. *Two point crossover*

Dua titik dipilih secara random untuk membagi orang tua yang pertama. Nilai gen di luar sisi dua titik yang terpilih, diturunkan ke kromosom anak, sesuai posisinya. Dari orang tua kedua, dilihat nilai gen yang lain, kemudian ditambahkan ke kromosom anak.

c. *Uniform crossover*

Pada *uniform crossover*, untuk memilih gen dari orang tua mana yang akan diturunkan ke kromosom anak dilakukan menggunakan rasio.

7. Mutasi

Mutasi merupakan proses pertukaran atau perubahan informasi yang dibawa oleh masing-masing gen dalam kromosom anak hasil pindah silang. Mutasi dilakukan pada beberapa gen dalam kromosom anak dengan probabilitas mutasi yang relatif kecil. Peluang mutasi (p_m) didefinisikan sebagai persentasi dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Apabila peluang mutasi terlalu besar maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya. Biasanya p_m sebesar $\frac{1}{n}$ dengan n adalah jumlah gen dalam kromosom.

Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Menurut Kusumadewi (2005: 247) mutasi terdapat 2 jenis yaitu mutasi biner dan mutasi bilangan real.

a. Mutasi biner

Cara sederhana untuk mendapatkan mutasi biner adalah dengan mengganti satu atau beberapa nilai gen dari kromosom. Langkah-langkah mutasi ini adalah sebagai berikut.

- 1) Jumlah gen pada populasi dihitung dengan cara panjang kromosom dikalikan dengan ukuran populasi.

- 2) Gen yang akan dimutasi dipilih secara acak.
- 3) Kromosom ditentukan berdasarkan gen yang terpilih untuk dimutasi.
- 4) Nilai gen diganti dari 0 ke 1 atau 1 ke 0 dari kromosom yang akan dimutasi tersebut.

Contoh mutasi biner dapat dilihat pada gambar 2.4.

kromosom sebelum mutasi	1	1	0	1	0	1	1	1	1	0
kromosom sesudah mutasi	1	1	0	1	0	0	1	1	1	0

Gambar 2.3 Contoh mutasi biner

b. Mutasi bilangan real

Pada mutasi bilangan real, ukuran langkah mutasi biasanya sangat sulit ditentukan. Ukuran kecil biasanya memberikan hasil yang baik yaitu berjalan cepat, namun terkadang ukuran yang lebih besar juga berjalan lebih cepat.

8. Pencarian lokal (*local search*)

Pencarian lokal (*local search*) merupakan pertukaran kembali informasi yang dibawa oleh gen-gen dalam satu kromosom anak. Pencarian lokal dapat dilakukan dengan menukar dua gen, atau permutasi beberapa gen tanpa mengurangi kualitas dari kromosom sebelumnya.

Pada algoritma memetika, tahapan pencarian lokal diterapkan pada 2 tempat yaitu sebelum masuk proses evolusi dan pada proses evolusi. Penjelasan mengenai tahapan tersebut adalah sebagai berikut.

- a. Sebelum masuk ke proses evolusi, pencarian lokal diterapkan terhadap semua kromosom pada populasi awal.
- b. Pada proses evolusi, pencarian lokal diterapkan sebelum atau sesudah proses seleksi, *crossover* atau mutasi namun tidak diterapkan untuk semua kromosom dalam populasi.

Menurut Rossidoria (2004: 3), dalam penjadwalan pencarian lokal yang efektif terdiri atas dua tahap yaitu tahap memperbaiki dan tahap meningkatkan kualitas jadwal. Penjelasan tahap tersebut adalah sebagai berikut.

- a. Tahap memperbaiki jadwal dari yang tidak layak menjadi layak adalah dengan mengurangi slot waktu yang digunakan.
- b. Tahap meningkatkan kualitas jadwal layak adalah dengan mengurangi pelanggaran pada kendala *soft*.

9. Pergantian populasi

Pergantian populasi merupakan pembentukan populasi baru yang beranggotakan kromosom-kromosom yang lebih berkualitas. Dengan pergantian populasi yang dilakukan terus menerus diharapkan kandidat solusi permasalahan semakin mendekati optimal. Pergantian populasi dilakukan dengan memasukkan satu atau dua kromosom anak ke populasi awal dan membuang satu atau dua kromosom dalam populasi awal yang mempunyai nilai penalti besar.

F. Aplikasi Algoritma Memetika pada Masalah Penjadwalan

Menurut Burke dkk (2004: 3), secara umum tugas dalam penjadwalan adalah mengakomodasi suatu hal misalnya kegiatan, kendaraan, dan sebagainya dalam pola

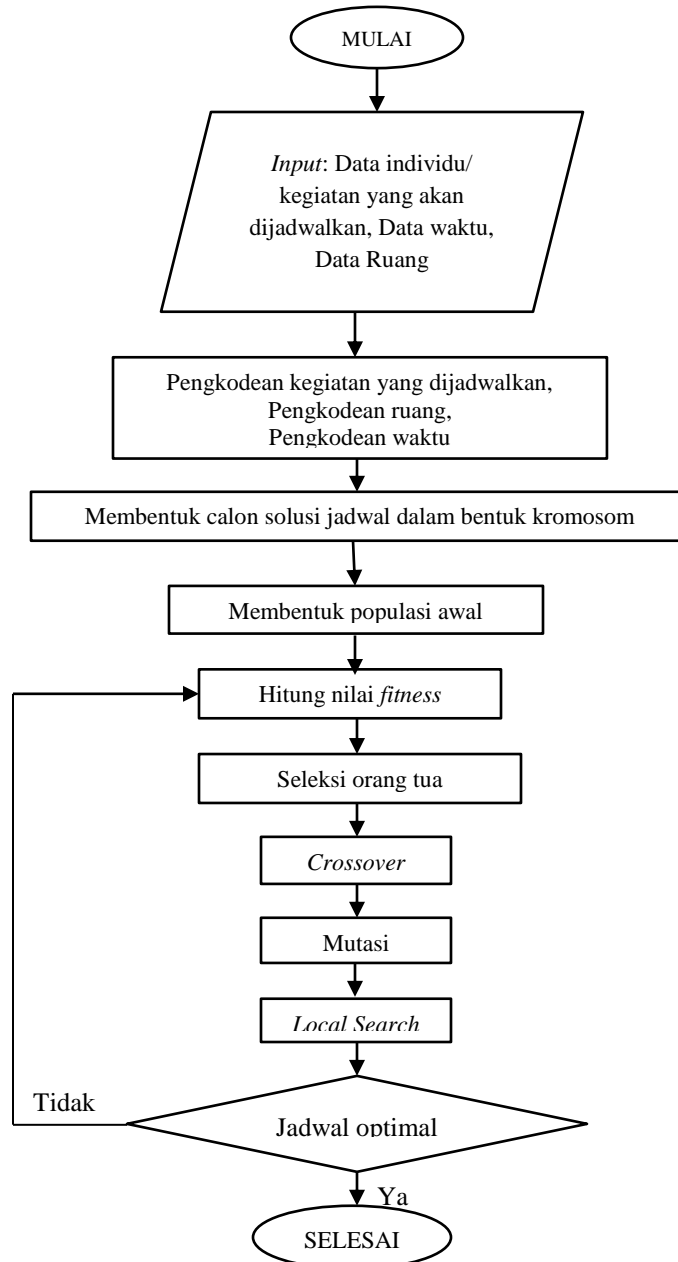
ruang dan waktu agar dapat memanfaatkan sumber daya yang tersedia dengan metode yang tepat sehingga kendala dapat terpenuhi. Masalah penjadwalan merupakan domain yang menarik untuk aplikasi algoritma memetika. Algoritma memetika menggunakan pencarian lokal yang berfungsi sebagai mekanisme efektif yang berfungsi ketika menggunakan representasi yang canggih. Aspek penting lainnya yang telah diteliti ketika merancang algoritma memetika untuk masalah penjadwalan adalah bagaimana membangun keseimbangan yang tepat antara langkah yang dilakukan algoritma genetika dan langkah yang dilakukan pencarian lokal.

Menurut Burke dkk (2004: 18), beberapa alasan algoritma memetika menjadi pendekatan yang baik dalam memecahkan berbagai masalah penjadwalan adalah sebagai berikut.

1. Ukuran ruang pencarian dalam permasalahan penjadwalan merupakan sebagian masalah dalam dunia nyata sehingga dibutuhkan kemampuan eksploratif yang baik seperti yang dimiliki algoritma memetika.
2. Masalah penjadwalan dibatasi oleh berbagai kendala sehingga seringkali lebih mudah untuk meningkatkan solusi *self-improve*. Sifat ini dibatasi oleh kendala yang mengarah pada pengkodean khusus dan operator untuk *self-improvement* yang didasarkan pada domain masalah.
3. Suatu fungsi evaluasi *fitness* dapat memakan waktu pada permasalahan penjadwalan. Penggunaan evaluasi perkiraan atau pendekatan dalam menentukan solusi pada algoritma memetika lebih kuat dibandingkan dengan metode *single-solution*. Hal ini dikarenakan algoritma memetika mempunyai satu populasi

dalam satu solusi baru yang dapat membantu mengurangi efek kesalahan dari pendekatan *fitness*.

Langkah-langkah proses penyusunan jadwal dengan algoritma memetika secara ringkas dapat digambarkan dalam diagram alir sebagai berikut.



Gambar 2.4 Diagram Alir Algoritma Memetika pada Proses Penjadwalan